

УДК 539.3: 519.6

ПАРАЛЛЕЛЬНОЕ РЕШЕНИЕ УПРУГОПЛАСТИЧЕСКОЙ ЗАДАЧИ С ПРИМЕНЕНИЕМ ТРЕХДИАГОНАЛЬНОГО АЛГОРИТМА LU-РАЗЛОЖЕНИЯ ИЗ БИБЛИОТЕКИ SCALAPACK

А.В. Коновалов, А.В. Толмачев, А.С. Партин

Институт машиноведения УрО РАН, Екатеринбург, Россия

Рассмотрено параллельное решение упругопластической задачи с применением трёхдиагонального алгоритма, основанного на LU-разложении матрицы системы линейных алгебраических уравнений, из библиотеки ScaLAPACK. Представлено краткое описание алгоритма и результаты вычислительных экспериментов.

Ключевые слова: упругопластическая задача, параллельные вычисления, ScaLAPACK

PARALLEL SOLUTION OF THE ELASTIC-PLASTIC PROBLEM USING A TRIDIAGONAL ALGORITHM FOR LU-DECOMPOSITION FROM THE SCALAPACK LIBRARY

A.V. Konovalov, A.V. Tolmachev and A.S. Partin

Institute of Engineering Science UB RAS, Ekaterinburg, Russia

The paper considers the parallel solution of the elastic-plastic problem using a tridiagonal LU-based solver from the ScaLAPACK library, including the solution of linear systems of equations. A brief description of the tridiagonal matrix algorithm is presented. The results of numerical experiments are given.

Key words: elastic-plastic problem, parallel computing, ScaLAPACK

1. Введение

Упругопластическая задача с большими пластическими деформациями физически и геометрически существенно нелинейна и требует большого количества времени для её решения на персональном компьютере. Например, на решение двумерной задачи методом конечных элементов (МКЭ) на регулярной сетке с одинаковым числом ячеек, равным 200 вдоль каждой из сторон расчетной области, затрачивается несколько часов; для трёхмерной задачи с аналогичной сеткой это время увеличивается до нескольких суток. Существенно сократить время вычислений можно с помощью техники с параллельной архитектурой, в частности, кластерных систем.

Реализация МКЭ в упругопластических задачах осуществляется в условиях пошагового нагружения и на каждом таком шаге состоит из трёх основных этапов [1]:

1) расчёт локальных матриц жёсткости для конечных элементов и формирование матрицы \mathbf{A} (глобальной матрицы жёсткости) и вектора \mathbf{b} правой части системы линейных алгебраических уравнений (СЛАУ)

$$\mathbf{Ax} = \mathbf{b} \quad (1)$$

относительно искомого вектора \mathbf{x} обобщённой скорости в узлах конечно-элементной сетки;

2) решение СЛАУ (1);

3) вычисление напряжённо-деформированного состояния конечных элементов в конце шага нагружения.

$$\left(\begin{array}{ccc|ccc} \mathbf{A}_1 & & & \mathbf{B}_1^U & & \\ & \mathbf{A}_2 & & \mathbf{D}_2^L & \mathbf{B}_2^U & \\ & & \ddots & & \ddots & \\ & & & & & \mathbf{B}_{p-1}^U \\ & & & & & \mathbf{D}_p^L \\ \hline & & & \mathbf{A}_p & & \\ \mathbf{B}_1^L & \mathbf{D}_2^U & & & & \\ & \mathbf{B}_2^L & & & & \\ & & \ddots & & & \\ & & & & & \mathbf{B}_{p-1}^L \\ & & & & & \mathbf{D}_p^U \end{array} \right) \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_{p-1} \\ \mathbf{x}_p \\ \vdots \\ \mathbf{x}_{p-1} \end{pmatrix} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_{p-1} \\ \mathbf{b}_p \\ \vdots \\ \mathbf{b}_{p-1} \end{pmatrix}, \text{ которую можно записать в более}$$

общей блочной форме

$$\begin{pmatrix} \hat{\mathbf{A}} & \mathbf{M}^U \\ \mathbf{M}^L & \mathbf{C} \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \xi \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \beta \end{pmatrix}. \quad (2)$$

LU-разложение [5] матрицы системы (2) имеет вид:

$$\begin{pmatrix} \hat{\mathbf{A}} & \mathbf{M}^U \\ \mathbf{M}^L & \mathbf{C} \end{pmatrix} = \begin{pmatrix} \mathbf{L} & \mathbf{0} \\ \mathbf{M}^L \mathbf{U}^{-1} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{U} & \mathbf{L}^{-1} \mathbf{M}^U \\ \mathbf{0} & \mathbf{S} \end{pmatrix}, \quad (3)$$

где \mathbf{L} и \mathbf{U} — соответственно нижняя и верхняя треугольные матрицы для матрицы $\hat{\mathbf{A}}$, $\mathbf{S} = \mathbf{C} - \mathbf{M}^L \mathbf{U}^{-1} \mathbf{L}^{-1} \mathbf{M}^U$ — дополнение Шура [5] для матрицы $\hat{\mathbf{A}}$, \mathbf{I} — единичная матрица.

Дополнение Шура \mathbf{S} для матрицы $\hat{\mathbf{A}}$ представляет собой блочную трёхдиагональную матрицу размерностью $(p-1) \times (p-1)$ с блоками $k \times k$:

$$\mathbf{S} = \begin{pmatrix} \mathbf{T}_1 & \mathbf{R}_2 & & & \\ \mathbf{V}_2 & \mathbf{T}_2 & \mathbf{R}_3 & & \\ & \ddots & \ddots & \ddots & \\ & & & \mathbf{R}_{p-1} & \\ & & & & \mathbf{T}_{p-1} \end{pmatrix}, \quad \begin{aligned} \mathbf{R}_i &= -\mathbf{D}_i^U \mathbf{U}_i^{-1} \mathbf{L}_i^{-1} \mathbf{B}_i^U, & \mathbf{V}_i &= -\mathbf{B}_i^L \mathbf{U}_i^{-1} \mathbf{L}_i^{-1} \mathbf{D}_i^L, \\ \mathbf{T}_i &= \mathbf{C}_i - \mathbf{B}_i^L \mathbf{U}_i^{-1} \mathbf{L}_i^{-1} \mathbf{B}_i^U - \mathbf{D}_{i+1}^U \mathbf{U}_{i+1}^{-1} \mathbf{L}_{i+1}^{-1} \mathbf{D}_{i+1}^L \quad (i=1, \dots, p-1). \end{aligned} \quad (4)$$

Решение системы (2) с учетом разложения (3) распадается на последовательное решение двух систем:

$$\begin{pmatrix} \mathbf{L} & \mathbf{0} \\ \mathbf{M}^L \mathbf{U}^{-1} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{c} \\ \gamma \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \beta \end{pmatrix}, \quad \begin{pmatrix} \mathbf{U} & \mathbf{L}^{-1} \mathbf{M}^U \\ \mathbf{0} & \mathbf{S} \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \xi \end{pmatrix} = \begin{pmatrix} \mathbf{c} \\ \gamma \end{pmatrix}. \quad (5)$$

В системах (5) \mathbf{c}_i и β_i — i -е части векторов \mathbf{c} и β , вычисляются по формулам $\mathbf{c}_i = \mathbf{L}_i^{-1} \mathbf{b}_i$, $\gamma_i = \beta_i - \mathbf{B}_i^L \mathbf{U}_i^{-1} \mathbf{c}_i - \mathbf{D}_{i+1}^U \mathbf{U}_{i+1}^{-1} \mathbf{c}_{i+1}$.

Каждый i -й процессор независимо от других процессоров вычисляет разложение блока диагонали $\mathbf{A}_i = \mathbf{L}_i \mathbf{U}_i$, а также произведения матриц $\mathbf{B}_i^L \mathbf{U}_i^{-1}$, $\mathbf{L}_i^{-1} \mathbf{B}_i^U$, $\mathbf{D}_i^U \mathbf{U}_i^{-1}$, $\mathbf{L}_i^{-1} \mathbf{D}_i^L$, $\mathbf{L}_i^{-1} \mathbf{b}_i$ и формирует i -ю часть матриц редуцированной системы $\mathbf{S} \xi = \gamma$: $\mathbf{C}_i - \mathbf{B}_i^L \mathbf{U}_i^{-1} \mathbf{L}_i^{-1} \mathbf{B}_i^U$, $\mathbf{D}_i^U \mathbf{U}_i^{-1} \mathbf{L}_i^{-1} \mathbf{D}_i^L$, $\mathbf{D}_i^U \mathbf{U}_i^{-1} \mathbf{L}_i^{-1} \mathbf{B}_i^U$, $\mathbf{B}_i^L \mathbf{U}_i^{-1} \mathbf{L}_i^{-1} \mathbf{D}_i^L$, $\beta_i - \mathbf{B}_i^L \mathbf{U}_i^{-1} \mathbf{c}_i$, $\mathbf{D}_i^U \mathbf{U}_i^{-1} \mathbf{c}_i$. Выполнение данных операций не требует межпроцессорных коммуникаций. Для экономии памяти матрицы $\mathbf{B}_i^L \mathbf{U}_i^{-1}$, $\mathbf{L}_i^{-1} \mathbf{B}_i^U$, $\mathbf{D}_i^U \mathbf{U}_i^{-1}$, $\mathbf{L}_i^{-1} \mathbf{D}_i^L$ и $\mathbf{L}_i^{-1} \mathbf{b}_i$ помещаются в те же места памяти, которые выделены для матриц \mathbf{B}_i^L , \mathbf{B}_i^U , \mathbf{D}_i^U и \mathbf{D}_i^L соответственно. Так как в процессе вычислений в матрицах \mathbf{D}_i^U и \mathbf{D}_i^L происходит дополнительное заполнение, то требуется дополнительное место в памяти для $2kn$ чисел с плавающей точкой. Поэтому общее требование к памяти примерно в два раза больше, чем для последовательного алгоритма. После выполнения локальных вычислений для завершения построения матрицы \mathbf{T}_i согласно (4) процессор с номером i посылает процессору с номером $i+1$ матрицу $\mathbf{C}_i - \mathbf{B}_i^L \mathbf{U}_i^{-1} \mathbf{L}_i^{-1} \mathbf{B}_i^U$.

Чётно-нечётная перестановка, вычисление разложения и формирование редуцированной системы повторяются до тех пор, пока матрица редуцированной системы не станет заполненной матрицей размерностью $k \times k$. В этом случае получившаяся система может быть решена на одном процессоре. Так как на каждой итерации редукации размерность матрицы уменьшается в два раза, то число шагов, необходимых для достижения желаемого результата, равно целой части от $\log_2(p-1)$.

После нахождения векторов ξ_i , каждый из процессоров вычисляет свою часть вектора \mathbf{x} : $\mathbf{x}_1 = \mathbf{U}_1^{-1}(\mathbf{c}_1 - \mathbf{L}_1^{-1} \mathbf{B}_1^U \xi_1)$, $\mathbf{x}_i = \mathbf{U}_i^{-1}(\mathbf{c}_i - \mathbf{L}_i^{-1} \mathbf{D}_i^L \xi_{i-1} - \mathbf{L}_i^{-1} \mathbf{B}_i^U \xi_i)$, $\mathbf{x}_p = \mathbf{U}_p^{-1}(\mathbf{c}_p - \mathbf{L}_{p-1}^{-1} \mathbf{D}_{p-1}^L \xi_{p-1})$.

3. Результаты вычислительных экспериментов

Все численные эксперименты проводились на решении методом конечных элементов осесимметричной задачи сжатия цилиндра плоскими плитами из упругопластического изотропного и изотропно-упрочняемого материала. В силу симметрии рассматривалась 1/4 часть продольного сечения цилиндра. Решение основывалось на принципе виртуальной мощности в скоростной форме [1]:

$$\int_V (\boldsymbol{\sigma} + \Delta t \dot{\boldsymbol{\sigma}}) \cdot \nabla \mathbf{h} dV + \int_{\Sigma} (\mathbf{P} + \Delta t \dot{\mathbf{P}}) \cdot \mathbf{h} d\Sigma = 0$$

со следующими определяющими соотношениями, полученными в работе [6]:

$$\begin{aligned} \dot{\boldsymbol{\sigma}} &= \lambda \dot{\Theta} \mathbf{I} + 2(\lambda \Theta + \mu) \mathbf{D} - \nabla \mathbf{v} \cdot \boldsymbol{\sigma} - \boldsymbol{\sigma} \cdot \nabla \mathbf{v}^T - J \omega \mathbf{S}, \\ \mathbf{S} &= \boldsymbol{\sigma} - \sigma_0 \mathbf{I}, \quad \sigma_0 = K \Theta, \quad K = \lambda + 2/3 \mu, \quad \mathbf{D} = 0,5(\nabla \mathbf{v} + \nabla \mathbf{v}^T), \\ \omega &= \left\{ \left[\mu \left(1 - \frac{2}{3} \Theta \right) \mathbf{S} - \mathbf{S} \cdot \mathbf{S} \right] \cdot \mathbf{D} - \frac{\mu}{3} \dot{\Theta} \mathbf{S} \cdot \mathbf{I} \right\} / \left[k^2 \left(1 + \frac{1}{\mu} \frac{dk}{d\chi} \right) \right], \\ F(\mathbf{S}, k) &= 0,5 \mathbf{S} \cdot \mathbf{S} - k^2 = 0 \quad \text{— условие текучести Мизеса.} \end{aligned} \tag{6}$$

Здесь: $\boldsymbol{\sigma}$ — тензор напряжений Коши; \mathbf{P} — плотность поверхностных сил; Δt — промежуток времени для шага приращения нагрузки; \mathbf{h} — вариация кинематически допустимых полей скоростей; ∇ — набла-оператор; V, Σ — объем и поверхность цилиндра и их элементы $dV, d\Sigma$ соответственно; λ, μ — коэффициенты Ламе; \mathbf{I} — единичный тензор; точкой и двумя точками обозначено, соответственно, скалярное и двойное скалярное произведение тензоров; точка над символом обозначает полную производную по времени; $\nabla \mathbf{v}$ — градиент скорости перемещений; \mathbf{D} — тензор скоростей деформаций; \mathbf{S} — девиатор тензора напряжений; σ_0 — среднее нормальное напряжение; K — объёмный модуль упругости; Θ — относительное изменение индивидуального объёма бесконечно малой частицы среды; k — напряжение текучести; J — числовой коэффициент, принимающий значения $J = 0$ (при $F < 0$ или при $F = 0, \mathbf{S} \cdot \dot{\mathbf{S}} \leq 0$) и $J = 1$ (при $F = 0, \mathbf{S} \cdot \dot{\mathbf{S}} > 0$); χ — параметр упрочнения (в силу малости упругих деформаций $\dot{\chi} = \sqrt{2 \mathbf{D} \cdot \mathbf{D}}$); ω — пластический модуль, который вычисляется итерационно для удовлетворения равенства (6) в конце каждого шага нагружения. Опыт вычислений показал, что данное условие с приемлемой точностью выполняется после 10–15 итераций.

На контакте с плитой предполагалось выполнение закона трения Кулона $\mathbf{T} = -\psi |\mathbf{P}_n| \mathbf{i}$, где \mathbf{T} и $|\mathbf{P}_n|$ — соответственно касательная составляющая и модуль нормальной составляющей вектора напряжений на контактной поверхности цилиндра; \mathbf{i} — единичный касательный вектор к контактной поверхности цилиндра, направленный вдоль его радиуса; ψ — коэффициент трения (в расчетах считался равным $\psi = 0,1$,

чтобы исключить, с целью упрощения алгоритма решения задачи, переход свободной поверхности на контакт с плитой). Нагружение в виде перемещения плиты осуществлялось ступенчато с шагом Δh . Шаг Δh выбирался так, чтобы отношение $\Delta h/h$ (h — высота цилиндра) не превышало предел упругости по деформации, в нашем случае — 0,002, что обеспечивало устойчивость вычислительной процедуры. Величина относительного сжатия цилиндра принималась равной 0,5.

Вычисления проводились на кластере um64 Института математики и механики УрО РАН, состоящем из 32 двухпроцессорных двухъядерных модулей на базе процессоров AMD Opteron с тактовой частотой 2,6 ГГц. Использовался интерфейс передачи сообщений из библиотеки OpenMPI версии 1.3.3 на сети InfiniBand. Параллельное решение СЛАУ по описанному выше алгоритму было осуществлено с помощью библиотеки ScaLAPACK, реализованной в Intel MKL [<http://software.intel.com/en-us/articles/intel-mkl/>] версии 10.0.010. Конечно-элементные сетки имели параметры, представленные в таблице 1.

Таблица 1. Параметры сеток в вычислительных экспериментах

Число ячеек по высоте цилиндра d	Полуширина ленты матрицы СЛАУ k	Число неизвестных n	k/n
70	145	10082	0,0144
100	205	20402	0,0100
120	245	29282	0,0084
150	305	45602	0,0067
200	405	80802	0,0050
300	605	181202	0,0033

На рисунке 2 приведены доли времени T , приходящиеся на вычисления отдельного этапа в пределах одного шага нагружения при решении задачи на одном процессоре с использованием метода LU-разложения для решения СЛАУ. Видно, что доля времени на решение СЛАУ увеличивается с ростом параметра разбиения d и на мелких сетках занимает почти половину общего времени расчётов на шаге нагружения.

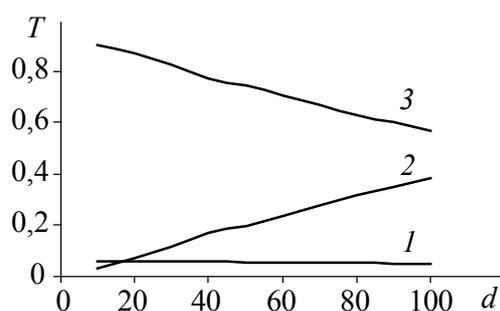


Рис. 2. Доли времени T осуществления этапов решения на одном шаге нагружения в задаче сжатия цилиндра: формирование матрицы жёсткости A (кривая 1), решение СЛАУ (2), вычисление напряжённо-деформированного состояния в конечных элементах в конце шага нагружения (3)

Рисунок 3 демонстрирует характер поведения и величину коэффициента ускорения a_p при распараллеливании различных этапов решения упругопластической задачи сжатия цилиндра. Здесь и далее коэффициент ускорения есть $a_p = t_1/t_p$, где t_1 и t_p — соответственно время выполнения последовательного алгоритма на одном процессоре и параллельного алгоритма на многопроцессорной вычислительной системе с числом процессоров $p > 1$.

Значения a_p при распараллеливании этапа формирования матрицы жёсткости A представлены на рисунке 3, а. На сетке с параметром разбиения $d = 300$ время формирования матрицы жёсткости с использованием только одного процессора получено путём экстраполяции результатов, так как для осуществления вычислений на одном узле вычислительной системы не хватало оперативной памяти процессора. Приведённые данные

свидетельствуют об эффективности распараллеливания этого этапа. Видно, что ускорение при увеличении количества процессоров практически не зависит от размера сетки.

Рисунок 3, б содержит зависимости коэффициента ускорения a_p от числа задействованных процессоров при распараллеливании этапа вычисления напряжённо-деформированного состояния конечных элементов в конце шага нагружения на p процессорах. Графики показывают хорошую эффективность распараллеливания данного этапа вычислений. Она выше, чем на этапе формирования матрицы жёсткости. Так же, как и на этапе формирования матрицы жёсткости, зависимость коэффициента ускорения от размеров сетки не наблюдается.

Коэффициент ускорения a_p , соответствующий этапу параллельных вычислений LU-разложения (5) матрицы \mathbf{A} системы (1), представлен на рисунке 3, в. Видно, что ускорение вычислений при распараллеливании появляется на сетках с параметром разбиения $d > 100$. При меньшем d происходит замедление вычислений за счёт того, что время на передачу данных превышает время, непосредственно затрачиваемое на вычисление на процессорах. Излом кривых при переходе на число процессоров больше двух получается из-за того, что объём вычислений в параллельном алгоритме примерно в 2 раза больше, чем в последовательном, а также прибавляются дополнительные затраты на передачу данных по сети. С увеличением количества разбиений уменьшается относительная ширина ленты матрицы \mathbf{A} , и коэффициент ускорения становится больше 1. Его величина тем больше, чем больше количество разбиений сетки, однако для сравнительно небольших сеток рост коэффициента ускорения замедляется с увеличением количества процессоров.

На рисунке 3, г приведены значения коэффициента ускорения a_p при распараллеливании решения системы (5). Видно, что при всех рассматриваемых значениях параметра разбиения сетки происходит замедление решения данной системы с ростом числа процессоров, и чем больше значение параметра сетки d , тем меньше скорость замедления.

В таблице 2 представлены значения времени, затрачиваемого на одно решение системы (5). Несмотря на то, что ускорение расчетов не наблюдается даже на сетках с параметром разбиения $d = 200$, время, затрачиваемое на решение системы при одном и том же числе процессоров, мало меняется с увеличением размера задачи (параметра разбиения сетки d). Это свидетельствует о том, что практически всё время, которое тратится на решение системы уравнений (5), определяется задержкой сети.

Таблица 2. Время, затрачиваемое на однократное решение системы (5)

Параметр разбиения d	Количество процессоров p						
	1	2	4	8	16	32	64
	Время, мс						
70	12,6	264,7	494,5	757,3	1112,1	1475,4	
100	33,8	281,4	539,3	762,2	1083,2	1526,5	
120	57,0	273,3	557,0	786,6	1138,3	2076,8	
150	127,3	331,2	569,4	787,6	1108,4	1495,1	1886,6
200	250,3	418,6	664,3	874,4	1183,7	1549,2	1919,9
300		832,7	1076,7	1231,2	1471,0	1794,2	2117,1

Рисунок 3, д демонстрирует зависимости коэффициента ускорения решения СЛАУ (1) при распараллеливании на шаге нагружения, когда используется LU-разложение матрицы системы \mathbf{A} и решение системы (5). До тех пор, пока не выполнилось условие пластичности, система (5) в рамках упругопластической задачи,

решалась 15 раз. При этом матрицы L и U оставались постоянными, а изменялась только правая часть матрицы B .

Видно, что при наличии ускорения для LU-разложения матрицы A (см. Рис.3, *а*) и его отсутствии при решении системы (5) (см. Рис. 3, *з*) в сумме ускорение проявляется на сетках с параметром разбиения $d = 200$. Это объясняется тем, что время решения системы (5) даже при большом количестве уравнений на несколько порядков меньше, чем время, затрачиваемое на LU-разложение матрицы, поэтому в итоге при распараллеливании вычислений выигрыш во времени наблюдается. Поскольку объём арифметических операций для LU-разложения есть асимптотически самая большая величина в решаемой задаче, то при дальнейшем увеличении количества разбиений сетки ускорение будет иметь место.

Полученные значения коэффициента ускорения a_p при распараллеливании решения всей упругопластической задачи в зависимости от количества используемых процессоров p и параметра разбиения сетки d показаны на рисунке 4. Видно, что для сеток с параметром $d > 150$ при количестве процессоров больше 4 коэффициент ускорения больше 1.

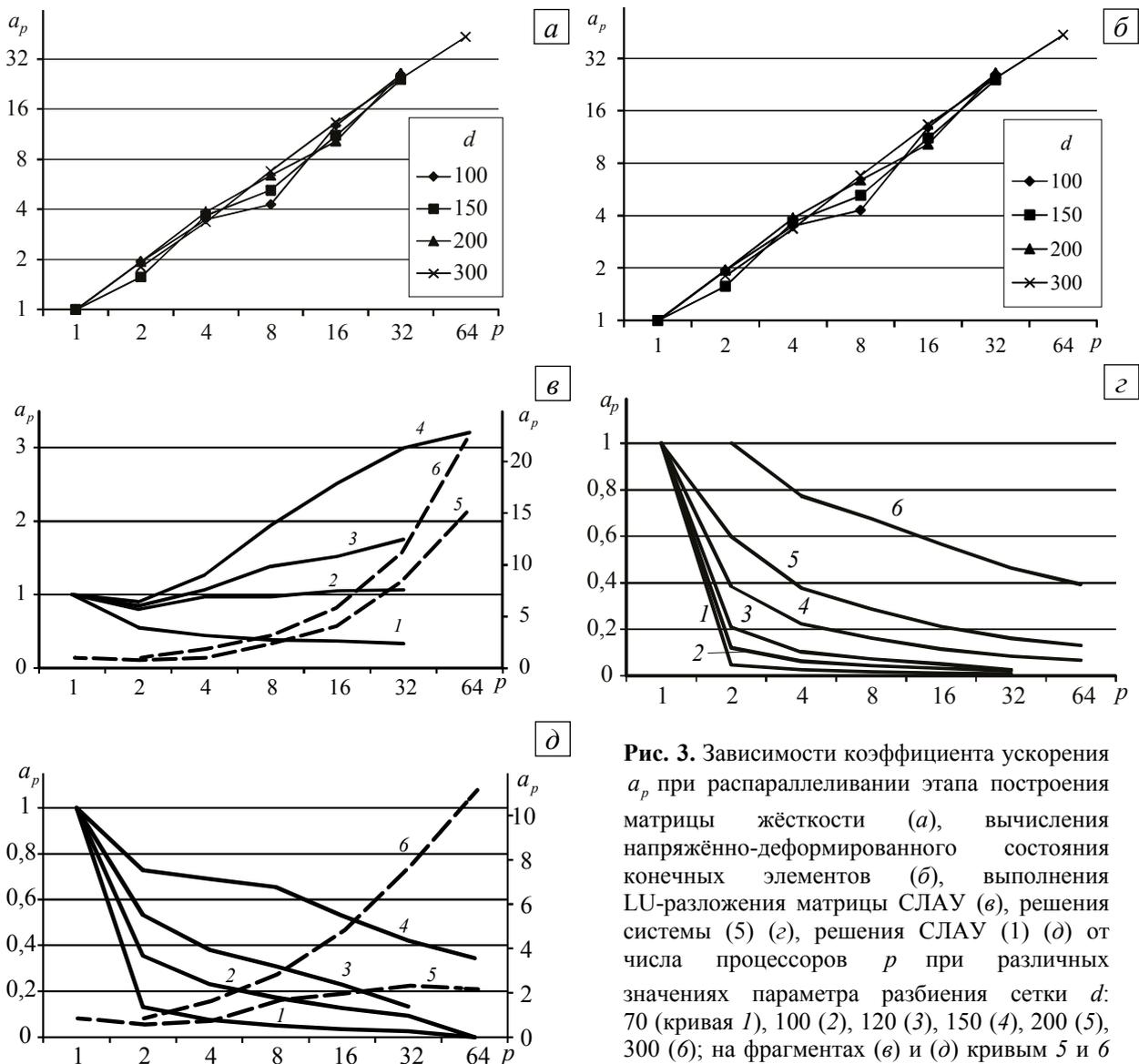


Рис. 3. Зависимости коэффициента ускорения a_p при распараллеливании этапа построения матрицы жёсткости (*а*), вычисления напряжённо-деформированного состояния конечных элементов (*б*), выполнения LU-разложения матрицы СЛАУ (*в*), решения системы (5) (*г*), решения СЛАУ (1) (*д*) от числа процессоров p при различных значениях параметра разбиения сетки d : 70 (кривая 1), 100 (2), 120 (3), 150 (4), 200 (5), 300 (6); на фрагментах (*в*) и (*д*) кривым 5 и 6 отвечает шкала справа

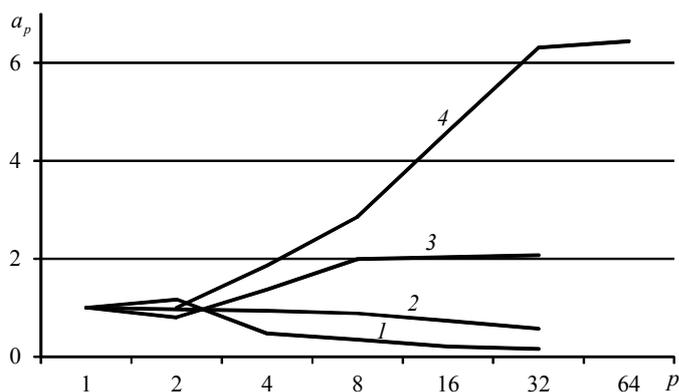


Рис. 4. Зависимости коэффициента ускорения a_p при распараллеливании всего решения упругопластической задачи от количества используемых процессоров p при различных значениях параметра разбиения сетки d : 100 (кривая 1), 150 (2), 200 (3), 300 (4)

Ускорение на двух процессорах для сетки с параметром разбиения $d = 100$ объясняется тем, что проигрыш во времени, связанный с вычислениями при использовании параллельного алгоритма LU-разложения, компенсируется выигрышем, полученным в результате распараллеливания расчёта напряженно-деформированного состояния. На сетке с параметром разбиения $d = 150$ эти значения компенсируют друг друга, поэтому ускорения не происходит. Затухание роста коэффициента ускорения, наблюдаемое на больших сетках, имеет место тогда, когда время, затрачиваемое на вычисление LU-разложения, становится одного порядка со временем, затрачиваемым на нахождение решения СЛАУ.

4. Заключение

При решении упругопластической задачи методом конечных элементов основное время затрачивается на решение СЛАУ. Так как матрица системы ленточная, и относительная ширина ленты уменьшается при увеличении количества разбиений конечно-элементной сетки, то рассмотренный в работе параллельный алгоритм решения СЛАУ можно использовать для решения упругопластических задач с параметрами регулярной сетки $k/n < 0,0067$.

Работа выполнена в рамках программы Президиума РАН «Интеллектуальные информационные технологии, математическое моделирование, системный анализ и автоматизация».

Литература

1. Поздеев А.А., Трусов П.В., Няшин Ю.И. Большие упруго-пластические деформации. – М.: Наука, 1986. – 232 с.
2. Cleary A., Dongarra J. Implementation in ScaLAPACK of divide-and-conquer algorithms for banded and tridiagonal linear systems: Computer Science Dept. Technical Report CS-97-358, 1997 – URL: <http://www.netlib.org/lapack/lawnpdf/lawn125.pdf> (дата обращения: 25.01.2011).
3. Arbenz P., Cleary A., Dongarra J., Hegland M.A. Comparison of parallel solvers for diagonally dominant and general narrow-banded linear systems // Tech. Report 312, ETH Zurich, Computer Science Dept, 1999 – URL: <http://www.netlib.org/lapack/lawnpdf/lawn142.pdf> (дата обращения: 25.01.2011).
4. Blackford L.S., Choi J., Cleary A., D'Azevedo E. et al. ScaLAPACK User's Guide. Philadelphia, USA, Society for Industrial and Applied Mathematics, 1997. – URL: <http://www.netlib.org/scalapack/slug> (дата обращения: 25.01.2011).
5. Кормен Т.Х., Лейзерсон Ч.И., Ривест Р.Л., Штайн К. Алгоритмы: построение и анализ. – М.: Вильямс, 2005. – 1296 с.
6. Коновалов А.В. Определяющие соотношения для упругопластической среды при больших пластических деформациях // Изв. РАН. МТТ. – 1997. – № 5. – С. 139-147.

Поступила в редакцию 28.02.11

Сведения об авторах

Коновалов Анатолий Владимирович, дтн, проф., зав.лаб., Институт Машиноведения УрО РАН (ИМАШ УрО РАН), 620049, г. Екатеринбург, ул. Комсомольская, д. 34; E-mail: avk@imach.uran.ru
 Толмачев Арсений Владимирович, асп., ИМАШ УрО РАН; E-mail: tolmachev.arseny@gmail.com
 Партин Александр Сергеевич, ктн, снс, ИМАШ УрО РАН; E-mail: lmd@imach.uran.ru